Mech. Sci., 13, 771–790, 2022
https://doi.org/10.5194/ms-13-771-2022

Mechanical Sciences
Open Access

TGA-based solutions map method for four-position synthesis of planar 4R linkage

Yehui Zhao[1,★], **Lijun Xue**[2,★], **Guangming Wang**[1,3,★], **Fanglei Zou**[4], **Yue Song**[5], **and Hongjian Zhang**[1,3]

[1]College of Mechanical and Electronic Engineering, Shandong Agricultural University, Taian, 271018, China
[2]College of Information Science and Engineering, Shandong Agricultural University, Taian, 271018, China
[3]Shandong Provincial Engineering Laboratory of Agricultural Equipment Intelligence, Taian, 271018, China
[4]College of Engineering, China Agricultural University, Beijing, 100083, China
[5]College of Engineering, Nanjing Agricultural University, Nanjing, 210031, China
★These authors contributed equally to this work.

Correspondence: Guangming Wang (gavinwang1986@163.com)

Received: 13 May 2022 – Revised: 3 August 2022 – Accepted: 25 August 2022 – Published: 28 September 2022

Abstract. In this study, the authors propose a *solutions map* based on a telomere genetic algorithm (TGA) to improve the efficiency of 4R-linkage synthesis. First, the points on the center curve are obtained by using projective geometry, and those of the circle curve are obtained by vector elimination. Second, the definition of the linkage type, assessment of linkage performance, and a method to identify defects in the linkage are introduced. Third, the solutions map method is proposed that can map the linkage solutions obtained by all combinations of the points on the center curve to a 3D color-coded surface to represent the distribution of solutions with different attributes of linkages in the solution domain. Fourth, we use the proposed *telomere operator* and *pseudo-histogram* method to improve the traditional genetic algorithm, and expand the domain of solutions of the solutions map by using the TGA. Finally, the linkage synthesis software *BurLink* is developed based on the solutions map method. The results show that the TGA-based solutions map can quickly locate the required 4R-linkage solution in the solution domain, and provides engineers with more candidate solutions than traditional methods.

1 Introduction

The positional synthesis of the 4R linkage is the inverse calculation of its kinematic analysis, i.e., to calculate the parameters of linkage according to several specified positions of the coupler. It is widely used in engineering. The Burmester theory (Cera and Pennestrì, 2018, 2019; Shirazi, 2007) points out that at most, five accurate positions can be specified in positional synthesis. Because the number of candidate solutions for a five-position synthesis problem is no more than six, this can easily lead to a situation where no viable solution is available, and this renders four-position synthesis more practical for use in engineering. According to the Burmester theory, any two center points yield a linkage solution such that the four-position synthesis problem has ∞^2 solutions in theory. Although there are many combinations of center points, most solutions involve defects in the circuit, branch, or order (Baskar and Bandyopadhyay, 2019; Singh et al., 2017; Tipparthi and Larochelle, 2011), which requires engineers to often spend weeks or even longer on repeated calculations and analyses to obtain satisfactory linkage solutions. Therefore, it is important to solve the problem of the poor efficiency of four-position synthesis. The goal of this study is to improve the efficiency of synthesis through a computer-aided method of synthesis. The theories of synthesis and analysis used here are introduced in Sects. 2 and 3, respectively, and the corresponding methods and software are provided in Sects. 4 and 5, respectively. The main conclusions are summarized in Sect. 6.

It is easy to determine that three factors affect the synthesis efficiency of the 4R linkage: (i) the lack of effective methods for identifying defective solutions, (ii) the blindness of the selection of points on the center curve, and (iii) the absence of a viable solution owing to the small number of available solutions in the domain. Previous studies have thus focused on these issues. To identify defective solutions, Filemon (1972) used the intersection of opposite sides of opposite pole quadrilaterals to divide the center curve into six segments, and found that the 4R linkage formed by the two center points in the same segment had the same order of couplers. Similarly, Waldron and Strong (1978), and Gupta and Beloiu (1998) used region segmentation technology with the Burmester curve to identify defects in the branch and circuit of the linkage. Based on the same technology, Sun and Waldron (1981) controlled the transmission angle of the linkage within a prescribed range while realizing defect-free linkage synthesis. The most effective method to mitigate the blindness of selection of points on the center curve involves visualizing the solution domain. McCarthy (2013) and Ruth and McCarthy (1999) proposed the *type map* method in their synthesis software *Sphinx* (the latest version is *SphinxPC*). It can build a mapping between the combination of points on the center curve and the attributes of linkage type. After *Sphinx*, the analysis of the solution domain has been extended gradually to other fields of linkage synthesis. In addition to the early software for linkage synthesis *Lincages* (Erdman and Loftness, 2005), Cao and Han (2020), Han and Cao (2018), and Liu and Han (2021) recently used solution-domain analysis to solve the problems of the synthesis of spatial linkages, such as RCCC, 1CS-4SS, and HCCC. In case no feasible solution is available, approximate motion synthesis expands the solution domain by introducing errors that can ensure the existence of the solution at the expense of accuracy. Many studies have contributed to this field. Gogate and Matekar (2012) used the evolutionary algorithm for the approximate motion synthesis of the 4R linkage and then applied this method to the synthesis of the Watt-I six-link mechanism (Gogate and Matekar, 2014). Sundram and Larochelle (2015) solved the mixed exact–approximate synthesis problem based on the constrained nonlinear optimization toolbox of MATLAB. Similarly, Ravani and Roth (1983) used kinematic mapping to transform the problem of approximate motion synthesis into that of curve fitting in the image space, while Ge et al. (2017) reduced this problem to that of finding a G-manifold that can best fit points in the image in the least-squares sense. Although previous studies in different fields have solved one of the above three problems that lead to poor efficiency of synthesis, they cannot solve all three at the same time. Integrating different methods may be an effective approach to solving all of these problems.

The region-based segmentation technology for the Burmester curve attempts to remove the points on the center curve or those in the circle curve that may cause defects before obtaining the linkage solution. However, this technology cannot accurately distinguish between defective and non-defective solutions, and excludes part of the reasonable linkage solutions while excluding defective solutions. This significantly reduces the chance of finding feasible solutions. By contrast, the type map method carries out defect discrimination once the parameters of linkage have been determined; thus, it has a higher accuracy than region-based segmentation technology. Moreover, type map can visualize the solution domain so that engineers can select the center points according to the type of linkage, thereby avoiding the blindness of the process of linkage synthesis. However, the number of candidate solutions provided by type map is usually too small to meet engineering needs. Approximate and accurate position synthesis are subjects in different fields of research. They can be used to solve the four-position synthesis problem and usually obtain only one linkage solution per calculation, but this solution is obtained by active search in the solution domain through an optimization algorithm. Theoretically, a satisfactory solution can be obtained directly by adding a sufficient number of constraints to the optimization model, but this is not feasible in many cases: a mathematical model with many constraints is too complex to establish, and strong constraints may also cause the optimization algorithm to fail to converge. To sum up, the type map method can completely replace the traditional synthesis method based on region segmentation technology. Therefore, only the possibility of its integration with the problem of approximate-motion synthesis needs to be considered. To solve the above three problems that affect the efficiency of synthesis, the concept of error in approximate-motion synthesis is introduced to accurate-position synthesis in this study to simultaneously expand and visualize the solution domain. Specifically, a solutions map method based on a telomere genetic algorithm (TGA) that can analyze and expand the solution domain is proposed in this study.

2 Position synthesis of 4R linkage

2.1 Generation of center curve

As shown in Fig. 1, the purpose of this study is to design a 4R linkage, $A_0A_1B_1B_0$, so that a point E on the coupler AB can pass through four specified positions $P_1 - - P_4$ in sequence. The specified positions of the coupler are defined in the planar rectangular coordinate system and denoted by P (x_P, y_P, δ), where x_P and y_P are the abscissa and the ordinate of point P, respectively, and δ is the angle of rotation of the coupler. According to the Burmester theory, the key to solving this problem is to calculate the curves of the center and circle. Because the center curve directly reflects the positions of the fixed hinge A_0 and B_0, it should be generated first.

For any two positions of the coupler P_i (x_{P_i}, y_{P_i}, δ_i) and P_j (x_{P_j}, y_{P_j}, δ_j), the pole point p_{ij} ($x_{\mathrm{p}_{ij}}$, $y_{\mathrm{p}_{ij}}$) can be calculated by the following equation:

$$\begin{cases} x_{\mathrm{p}ij} = \frac{x_{\mathrm{P}_i}+x_{\mathrm{P}_j}}{2} + \frac{y_{\mathrm{P}_i}-y_{\mathrm{P}_j}}{2}\cot\left(\frac{\delta_j-\delta_i}{2}\right) \\ y_{\mathrm{p}ij} = \frac{y_{\mathrm{P}_i}+y_{\mathrm{P}_j}}{2} - \frac{x_{\mathrm{P}_i}-x_{\mathrm{P}_j}}{2}\cot\left(\frac{\delta_j-\delta_i}{2}\right) \end{cases}, \quad (1)$$

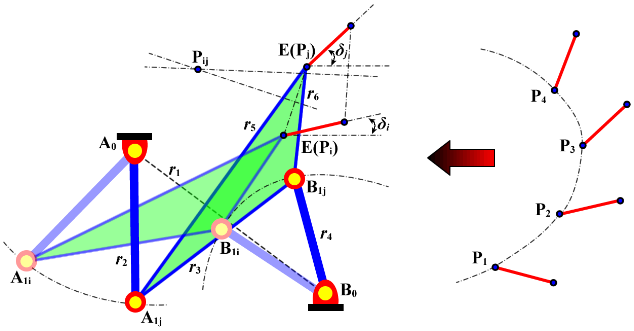


Figure 1. Motion of planar 4R linkage.

when the origin of the coordinate system moves to pole point p_{12}, the equation of the Burmester center curve C^{1234} can be expressed as follows (Zhang, 1983):

$$\begin{aligned}\left(x_{A0}^2+y_{A0}^2\right)(j_2x_{A0}-j_1y_{A0})+(j_1k_2-j_2k_1-j_3)x_{A0}^2\\ +(j_1k_2-j_2k_1+j_3)y_{A0}^2+2j_4x_{A0}y_{A0}\\ +(-j_1k_3+j_2k_4+j_3k_1-j_4k_2)x_{A0}\\ +(j_1k_4+j_2k_3-j_3k_2-j_4k_1)y_{A0}=0,\end{aligned} \tag{2}$$

where

$$\begin{aligned}&k_1=x_{\mathrm{p}13}+x_{\mathrm{p}24};\quad k_2=y_{\mathrm{p}13}+y_{\mathrm{p}24};\\ &k_3=x_{\mathrm{p}13}y_{\mathrm{p}24}+y_{\mathrm{p}13}x_{\mathrm{p}24};\quad k_4=x_{\mathrm{p}13}x_{\mathrm{p}24}-y_{\mathrm{p}13}y_{\mathrm{p}24};\\ &j_1=x_{\mathrm{p}23}+x_{\mathrm{p}14}-k_1;\quad j_2=y_{\mathrm{p}23}+y_{\mathrm{p}14}-k_2;\\ &j_3=x_{\mathrm{p}23}y_{\mathrm{p}14}+x_{\mathrm{p}13}y_{\mathrm{p}23}-k_3;\\ &j_4=x_{\mathrm{p}23}x_{\mathrm{p}14}-y_{\mathrm{p}23}y_{\mathrm{p}14}-k_4.\end{aligned} \tag{3}$$

The center curve C^{1234} is a third-order circular point curve. According to projective geometry (Jin and Shi, 1991), the curve usually has a real focus, and the solution of Eq. (2) can then be expressed as follows:

$$\begin{cases} x_{A0}=-\frac{\gamma}{2}\left[1\pm\sqrt{1-4(d+2c\sigma)\left(\gamma^2+4\sigma^2\right)}\right]\\ y_{A0}=\sigma\left[1\pm\sqrt{1-4(d+2c\sigma)\left(\gamma^2+4\sigma^2\right)}\right]\end{cases}, \tag{4}$$

where σ is the input variable, and the other variables are defined as:

$$\begin{aligned}&c=-e/\gamma;\, d=(2j_2b_4-2j_1b_5)/\left(j_1^2+j_2^2\right);\\ &e=(2j_2b_5+2j_1b_4)/\left(j_1^2+j_2^2\right);\\ &\gamma=\pm(3j_2x_F-j_1y_F+a_1)/\sqrt{j_1^2+j_2^2};\\ &b_4=j_2\left(x_F^2+y_F^2\right)/2+x_F(j_2x_F-j_1y_F)+a_1x_F+a_2y_F+a_4;\\ &b_5=-j_1\left(x_F^2+y_F^2\right)/2+y_F(j_2x_F-j_1y_F)+a_2x_F+a_3y_F+a_5;\\ &a_1=j_1k_2-j_2k_1-j_3;\\ &a_2=j_4;\\ &a_3=j_1k_2-j_2k_1+j_3;\\ &a_4=(-j_1k_3+j_2k_4+j_3k_1-j_4k_2)/2;\\ &a_5=(j_1k_4+j_2k_3-j_3k_2-j_4k_1)/2;\\ &x_F=(j_1j_4+j_2j_3)/\left(j_1^2+j_2^2\right);\\ &y_F=(j_1j_3-j_2j_4)/\left(j_1^2+j_2^2\right).\end{aligned} \tag{5}$$

Note that the sign of γ is "+" only when $j_2\geq 0$.

In this case, the shape of the center curve depends on the value of τ:

$$\tau=\frac{4\gamma^2d-\gamma^4+4e^2}{4\gamma^2}. \tag{6}$$

If $\tau>0$, C^{1234} is a non-closed curve, if $\tau=0$, C^{1234} is a non-closed curve with one coincident point, and if $\tau<0$, C^{1234} is divided into a non-closed curve and an elliptic curve. To control the distribution density of the generated Burmester center points, the input variable σ can be parameterized as follows:

$$\sigma=\alpha\tan^{(2k-1)}(\varphi)\quad \varphi\in\left[-\tfrac{\pi}{2},\tfrac{\pi}{2}\right)\cup\left(\tfrac{\pi}{2},\tfrac{3\pi}{2}\right], \tag{7}$$

where α and k are constants.

Note that the order of $\tan(\varphi)$ cannot be even. This leads to $\sigma/\alpha>0$, such that one part of the center curve is repeatedly calculated and the other part is lost.

To facilitate programming, the range of values of φ under any condition is set to $-\pi/2\sim 3\pi/2$ in this study. The equations based on this are shown in Table 1, and are used to generate each part of the center curve in turn to ensure that points of each part of the curve are connected from head to tail without jumps. Once all points on the center curve have been calculated, it is necessary to preprocess each point. For example, points on the center curve outside the set area are eliminated according to their coordinates, and center points with too small a spacing between them are eliminated according to the Euclidean distance between adjacent points. Three typical center curves generated by a computer program based on the above method are shown in Fig. 2.

Table 1. Generation of center curve C^{1234}.

τ	Generation order	Range of φ	Coordinates of C^{1234}
$\tau>0$	I	$\left(\frac{\pi}{2}, \arctan\left(-\frac{e}{\gamma}+\sqrt{\tau}\right)\right)$	$\begin{cases} x_{A0}=-\frac{\gamma}{2}\left[1+\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \\ y_{A0}=\sigma\left[1+\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \end{cases}$
	II	$\left(\arctan\left(-\frac{e}{\gamma}+\sqrt{\tau}\right)+\pi, \frac{3\pi}{2}\right)$	$\begin{cases} x_{A0}=-\frac{\gamma}{2}\left[1-\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \\ y_{A0}=\sigma\left[1-\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \end{cases}$
	III	$\left(-\frac{\pi}{2}, \arctan\left(-\frac{e}{\gamma}-\sqrt{\tau}\right)\right)$	$\begin{cases} x_{A0}=-\frac{\gamma}{2}\left[1-\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \\ y_{A0}=\sigma\left[1-\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \end{cases}$
	IV	$\left(\arctan\left(-\frac{e}{\gamma}-\sqrt{\tau}\right)+\pi, \frac{\pi}{2}\right)$	$\begin{cases} x_{A0}=-\frac{\gamma}{2}\left[1+\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \\ y_{A0}=\sigma\left[1+\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \end{cases}$
$\tau=0$	I	$\left(\frac{\pi}{2}, \arctan\left(-\frac{e}{\gamma}\right)\right)$	$\begin{cases} x_{A0}=-\frac{\gamma}{2}\left[1+\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \\ y_{A0}=\sigma\left[1+\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \end{cases}$
	II	$\left(\arctan\left(-\frac{e}{\gamma}\right), -\frac{\pi}{2}\right)$	$\begin{cases} x_{A0}=-\frac{\gamma}{2}\left[1-\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \\ y_{A0}=\sigma\left[1-\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \end{cases}$
	III	$\left(\frac{3\pi}{2}, \arctan\left(-\frac{e}{\gamma}\right)+\pi\right)$	$\begin{cases} x_{A0}=-\frac{\gamma}{2}\left[1-\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \\ y_{A0}=\sigma\left[1-\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \end{cases}$
	IV	$\left(\arctan\left(-\frac{e}{\gamma}\right)+\pi, \frac{\pi}{2}\right)$	$\begin{cases} x_{A0}=-\frac{\gamma}{2}\left[1+\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \\ y_{A0}=\sigma\left[1+\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \end{cases}$
$\tau<0$	I	$\left(\frac{\pi}{2}, -\frac{\pi}{2}\right)$	$\begin{cases} x_{A0}=-\frac{\gamma}{2}\left[1+\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \\ y_{A0}=\sigma\left[1+\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \end{cases}$
	II	$\left(\frac{3\pi}{2}, \frac{\pi}{2}\right)$	$\begin{cases} x_{A0}=-\frac{\gamma}{2}\left[1-\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \\ y_{A0}=\sigma\left[1-\sqrt{1-4(d+2c\sigma)(\gamma^2+4\sigma^2)}\right] \end{cases}$

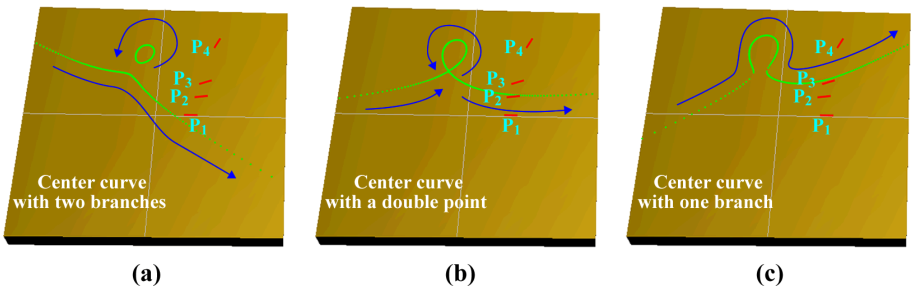


Figure 2. Center curves generated by a computer. **(a)** $\tau<0$. **(b)** $\tau=0$. **(c)** $\tau>0$.

2.2 Computation of points on the circle curve

Considering that it is difficult to generate the circle curve from the center curve by using the geometric method, the traditional vector elimination method (Liang and Chen, 1993) is improved so that it can generate the circle curve according to the coordinates of the points on the center curve. According to this method, the RR links (i.e., $A_0A_1 - A_1E$ in Fig. 1) satisfy the following equation in motion:

$$\begin{aligned} & r_5\cos\delta_0\left[\left(x_{A0}-x_{\mathrm{P}_i}\right)\cos\Theta\delta_i - y_{\mathrm{P}_i}\sin\Delta\delta_i\right] \\ & \quad + r_5\sin\delta_0\left[\left(x_{\mathrm{P}_i}-x_{A0}\right)\sin\Delta\delta_i - y_{\mathrm{P}_i}\cos\Delta\delta_i\right] \\ & \quad - y_{\mathrm{P}_i}y_{A0} + \frac{1}{2}\left(y_{A0}^2 - r_2^2 + r_5^2\right) \\ & \quad + y_{A0}r_5\sin\delta_0\cos\Delta\delta_i + y_{A0}r_5\cos\delta_0\sin\Delta\delta_i \\ & \quad + \left[\frac{1}{2}\left(x_{\mathrm{P}_i}^2 + y_{\mathrm{P}_i}^2 + x_{A0}^2\right) - x_{A0}x_{Pi}\right] = 0, \end{aligned} \tag{8}$$

where δ_0 is the initial angle of the link A_1E and $\Delta\delta_i = \delta_i - \delta_1$. We define the following:

$$\begin{aligned} & f_{0i} = \left(x_{A0}-x_{\mathrm{P}_i}\right)\cos\Delta\delta_i - y_{\mathrm{P}_i}\sin\Delta\delta_i; \\ & f_{1i} = -\left(x_{A0}-x_{\mathrm{P}_i}\right)\sin\Delta\delta_i - y_{\mathrm{P}_i}\cos\Delta\delta_i; \\ & f_{2i} = -y_{\mathrm{P}_i};\ f_{3i} = 1;\ f_{4i} = \cos\Delta\delta_i;\ f_{5i} = \sin\Delta\delta_i; \\ & F_i = \frac{1}{2}\left(x_{\mathrm{P}_i}^2 + y_{\mathrm{P}_i}^2 + x_{A0}^2\right) - x_{A0}x_{\mathrm{P}_i}; \\ & \vartheta_0 = r_5\cos\delta_0;\ \vartheta_1 = r_5\sin\delta_0; \\ & \vartheta_2 = \frac{1}{2}\left(y_{A0}^2 - r_2^2 + r_5^2\right);\ \vartheta_3 = y_{A0}; \\ & \vartheta_4 = y_{A0}r_5\sin\delta_0;\ \vartheta_5 = y_{A0}r_5\cos\delta_0; \\ & \vartheta_4 = \vartheta_1\vartheta_3;\ \vartheta_5 = \vartheta_0\vartheta_3. \end{aligned} \tag{9}$$

By substituting the above formulae into Eq. (8), the following equation is obtained after simplification:

$$\begin{bmatrix} f_{31} & f_{21} & f_{41} & f_{51} \\ f_{32} & f_{22} & f_{42} & f_{52} \\ f_{33} & f_{23} & f_{43} & f_{53} \\ f_{34} & f_{24} & f_{44} & f_{54} \end{bmatrix} \begin{bmatrix} \vartheta_2 \\ \vartheta_3 \\ \vartheta_4 \\ \vartheta_5 \end{bmatrix} = \begin{bmatrix} -f_{01} & -f_{11} & -F_1 \\ -f_{02} & -f_{12} & -F_2 \\ -f_{03} & -f_{13} & -F_3 \\ -f_{04} & -f_{14} & -F_4 \end{bmatrix} \begin{bmatrix} \vartheta_0 \\ \vartheta_1 \\ 1. \end{bmatrix} \tag{10}$$

After the numerical computation of matrix inversion, the above equation is transformed into

$$\begin{bmatrix} \vartheta_2 \\ \vartheta_3 \\ \vartheta_4 \\ \vartheta_5 \end{bmatrix} = \begin{bmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \\ A_4 & B_4 & C_4, \end{bmatrix} \begin{bmatrix} \vartheta_0 \\ \vartheta_1 \\ 1, \end{bmatrix} \tag{11}$$

where A_k, B_k, and C_k are elements of the matrix.

In previous research (Liang and Chen, 1993; Zhao, 2009), Eq. (11) has been decomposed into several sub-formulae that can be merged into a high-order equation with only one variable by elimination:

$$E_1\vartheta_1^3 + E_2\vartheta_1^2 + E_3\vartheta_1 + E_4 = 0, \tag{12}$$

where $E_1 - E_4$ are coefficients related to A_k, B_k, and C_k.

After solving the equation by dichotomy or Newton's iterative method, the values of all variables can be obtained in a step-by-step manner based on the relationships between them. Note that because it is difficult to unify the symbolic definition in Fig. 1 and the algebraic definition in Eq. (9), there are significant differences between Eqs. (11) and (12) as derived in the literature, but the principle of vector elimination considered is consistent. Although this method can offer a direct solution for the coordinates of points on the center curve and the circle curve, rectangular coordinates that cannot reflect the trend of the Burmester curve are used to generate the points on the center curve. For example, this method needs to first specify the abscissa of the point on the center curve, where this coordinate can be taken arbitrarily on the plane and its value is independent of the shape of the center curve. It is thus difficult to parameterize the calculation process and avoid jumps when the point on the center curve is generated. This is also why we use the projective geometry method to calculate the coordinates of the point on the center curve. As the center curve has been generated, the vector elimination method is modified in this study. The improved solution process is as follows (Eqs. 13–16):

The following equation is equivalent to Eq. (11):

$$\begin{cases} \vartheta_2 & = A_1\vartheta_0 + B_1\vartheta_1 + C_1 \\ \vartheta_4 & = A_3\vartheta_0 + B_3\vartheta_1 + C_3 = \vartheta_1\vartheta_3 \\ \vartheta_5 & = A_4\vartheta_0 + B_4\vartheta_1 + C_4 = \vartheta_0\vartheta_3, \end{cases} \tag{13}$$

According to the second and third sub-formulae of Eq. (13), the expressions containing ϑ_0 and ϑ_1 are obtained as follows:

$$\begin{cases} (B_3-\vartheta_3)\vartheta_1 + A_3\vartheta_0 + C_3 = 0 \\ B_4\vartheta_1 + (A_4-\vartheta_3)\vartheta_0 + C_4 = 0, \end{cases} \tag{14}$$

Further, ϑ_0 and ϑ_1 are obtained as follows:

$$\begin{cases} \vartheta_0 = \frac{C_4(B_3-\vartheta_3)-C_3B_4}{A_3B_4-(A_4-\vartheta_3)(B_3-\vartheta_3)} \\ \vartheta_1 = \frac{(\vartheta_3-A_4)\vartheta_0-C_4}{B_4}. \end{cases} \tag{15}$$

Once the coordinates of point $A_0(x_{A0},\ y_{A0})$, i.e., the Burmester center point, have been determined, ϑ_0, ϑ_1, and ϑ_3 ($= y_{A0}$, see Eq. 9) can be solved. By substituting ϑ_0 and ϑ_1 into the expressions of Eq. (9), r_5 and r_2 can be written as follows:

$$\begin{cases} r_5 = \sqrt{\vartheta_0^2 + \vartheta_1^2} \\ r_2 = \sqrt{r_5^2 + y_{A0}^2 - 2\vartheta_2} \end{cases}, \tag{16}$$

where ϑ_2 is given by the first sub-formula of Eq. (13).

Table 2. Classification of 4R linkages.

No.	T_1	T_2	T_3	Linkage type
1	−	−	−	Crank rocker
2	+	+	−	Rocker crank
3	+	−	+	Double crank
4	−	+	+	Grashof double rocker
5	+	+	+	0–0 double rocker
6	−	+	−	0–π double rocker
7	+	−	−	$\pi - 0$ double rocker
8	−	−	+	$\pi - \pi$ double rocker

When the coordinates of points A_0 and P_i as well as the link lengths r_2 and r_5 are known, the coordinates of the point in the circle curve at position i ($i = 1,2,3,4$) can be calculated as A_{1_i} (x_{A_0}+uw$_1$−vw$_2$, y_{A_0}+uw$_2$+vw$_1$) (Liang and Chen, 1993), where

$$w_1 = x_{\mathrm{P}_i} - x_{A0},\ w_2 = y_{\mathrm{P}_i} - y_{A0},\ w_3 = w_1^2 + w_2^2,\ u = \frac{w_3 + r_2^2 - r_5^2}{2w_3},\ v = \pm\sqrt{\frac{r_2^2}{w_3} - u^2}. \tag{17}$$

Note that the variable v has two solutions. For the correct solution, the rotational angle of the link A_1E should be equal to $\delta_0 + \Delta\delta_i$. The solution process does not involve the solution of the higher-order equation.

3 Analysis of solutions

3.1 Linkage type classification

According to the method proposed by Martin and Murray (2002), the 4R linkage can be classified into eight types based on the signs of T_1, T_2, and T_3, as shown in Table 2, where

$$\begin{cases} T_1 = |r_2| + |r_3| - |r_1| - |r_4| = 0 \\ T_2 = |r_2| - |r_3| + |r_1| - |r_4| = 0 \\ T_3 = |r_2| - |r_3| - |r_1| + |r_4| = 0. \end{cases} \tag{18}$$

3.2 Performance evaluation of linkage

As a discontinuous attribute, the linkage type is often not the only objective of linkage synthesis. It is thus also necessary to consider the values of continuous attributes, such as the link length, transmission angle, acceleration, and force, or their weighted values in the design of the linkage mechanism (Jia et al., 2021; Trejo et al., 2015; Wilhelm et al., 2017). For convenience of understanding, this study takes the transmission angle as the index to assess the performance of the linkage. As shown in Fig. 3, if the mass of each link and the friction of the revolute joint are neglected, the coupler A_1B_1 becomes a two-force link, and the acute angle γ between links A_1B_1 and B_0B_1 is the transmission angle. The larger the angle γ is, the greater is the force F_t of the driving force F in the direction of its velocity, and the better is the transmission performance of the 4R linkage. Therefore, when the coupler is in the four given positions, its minimum transmission angle $\gamma_{\min}$ can be used as an index for the evaluation of the transmission-related performance of the 4R linkage.

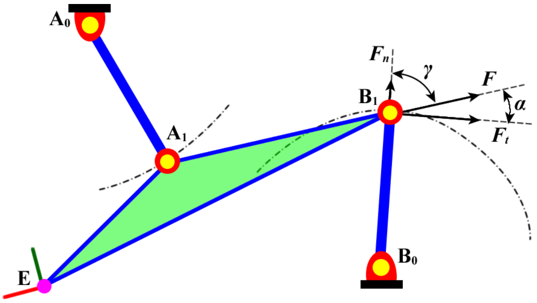


Figure 3. Transmission angle of the 4R linkage.

3.3 Defect discrimination

For each 4R linkage synthesized according to the Burmester theory, we need to discriminate among three kinds of defects, i.e., defects in the circuit, branch, and order, as shown in Fig. 4.

1. *Identifying circuit defects.* If the driving link A_0A_i is not fully rotatable, its range of motion is divided into n sectors based on the limited positions; see Fig. 4a. If the rotation angles of the driving link corresponding to the specified positions of the coupler are in different sectors, the 4R linkage has a circuit defect. In this case, the angles of rotation of the driving link at the limited positions (θ_2^*) and the specified positions (θ_{2_i}) should be calculated:

   $$\theta_2^* = \theta_1 \pm \arccos\left(\frac{r_1^2 + r_2^2 - (r_4 \pm r_3)^2}{2r_1r_2}\right), \tag{19}$$

   and

   $$\theta_{2_i} = \arctan 2\left(y_{A1i} - y_{A0i}, x_{A1i} - x_{A0i}\right). \tag{20}$$

   Because the domain of definition of the arccosine function is $[-1, 1]$, the number of solutions of θ_2^* in Eq. (19) may be zero, two, or four. There will be no circuit defect in the linkage if all four angles θ_{2_i} are located in the same sector defined by θ_2^*.

2. *Identifying branch defects.* As shown in Fig. 4b, the coupler of the 4R linkage usually has two forms of assembly, i.e., $A_0A_1B_{1(1)}B_0$ and $A_0A_1B_{1(2)}B_0$. It is difficult for the coupler of the 4R linkage to move from one form to another without jamming. If the planar 4R linkage is

placed in the X–Y plane of the 3D coordinate system O-XYZ, the following equation can be used to identify branch defects:

$$V_i\,(x_v y_v z_v) = B_0 B_{1i} \times B_{1i} A_{1i}. \tag{21}$$

Only if all the four components of the z axis of V_i (i.e., z_v) have the same sign ("+" or "−") when the driving link is located at the specified positions of the coupler, the linkage solution is valid such that it does not contain a branch defect.

3. *Identifying order defects.* If the driving link of the 4R linkage cannot pass through the four specified positions in the order "1-2-3-4," it has an order defect. If the driving link is fully rotatable, the order "1-2-3-4" is equivalent to the orders "2-3-4-1," "3-4-1-2," "4-1-2-3," "4-3-2-1," "3-2-1-4," "2-1-4-3," and "1-4-3-2." The defect-free linkage satisfies $\varphi_{12}<\varphi_{13}<\varphi_{14}$ or $\varphi_{12}>\varphi_{13}>\varphi_{14}$, where φ_{1_i} is the angle of rotation of the driving link when it rotates from position 1 to position i in the anti-clockwise direction. If the driving link is partly rotatable, see Fig. 4c, only "1-2-3-4" and "4-3-2-1" are valid orders because the driving link cannot move from one circuit to another. In this case, the rotational angles of the driving link need to satisfy the condition $\varphi_{t2}<\varphi_{t3}<\varphi_{t4}$ or $\varphi_{t2}>\varphi_{t3}>\varphi_{t4}$, where φ_{ti} is the angle of rotation of the driving link when it rotates from any limited position to the specified position i in the anti-clockwise direction.

4. *Organizational rules.* When circuit defects occur, the coupler link cannot pass through any of the four specified positions in any condition; when branch defects occur, it cannot pass through the four specified positions without external intervention; when order defects occur, the coupler link can pass through four specified positions but in the wrong order. Therefore, circuit defects are more severe than branch defects, which are more severe than order defects. To reduce the amount of requisite calculation, we stipulate that if a linkage has circuit defects, the programs no longer continue to identify its branch and order defects, and if the linkage has branch defects, the programs no longer continue to identify its order defects.

4 Basic solutions map method

4.1 Principle of solutions map method

The principle of the solutions map proposed in this paper is to map all linkage solutions to a 3D color-coded surface based on linkage type to visualize the distribution of the solutions. There are three steps to generate the solutions map:

1. *Generation of Burmester curve.* By using projective geometry, N center points can be obtained when the parameter φ changes from $-\pi/2$ to $3\pi/2$ (Sect. 2.1). Note that the number of center points N^* is usually less than N because of the preprocessing of the center points. For example, center points beyond the designated area or those with spacing that is too small between them should be excluded. Of course, designers can also adjust the distribution of the center points by changing the value of α or k in Eq. (7). After pretreatment, each center point is assigned a unique index (from 1 to N^*) for subsequent calculation. Finally, for each center point, four circle points corresponding to the four specified positions are calculated (Sect. 2.2).

2. *Generation of type map.* According to the Burmester theory, a solution of the 4R linkage for four specified positions can be determined by any two center points and the corresponding circle points. To describe the distribution of the solution, the x axis and y axis of the rectangular coordinate system are both divided into N^* parts to form a grid array of size $N^* \times N^*$. If i and $j \epsilon N^*$ are defined, the grid coordinates (i, j) represent the linkage solution obtained by combining the center points i and j. For each linkage solution, it is necessary to calculate the linkage type (Sect. 3.1) and fill the grid (i, j) with the color corresponding to it, as shown in Table 3.

3. *Generation of solutions map.* In total, nine linkage types can be color-coded when the types of defective linkages are considered. To describe the continuously changing attributes of the linkages, type map can be placed in the X–Y plane of the 3D coordinate system. Then, the z-axis of the coordinate system can be used to express the attributes of the linkage introduced in Sect. 3.2. For example, given the task of position synthesis shown in Table 4, the solutions map generated according to the above steps is shown in Fig. 5.

   The solutions map consists of the following parts:

   a. *Mountain range.* The surface of the solutions map is shaped like a mountain, and any point on the mountain corresponds to a synthesis-based solution of the 4R linkage for the four specified coupler positions. In addition, the color and height of points on the mountain represent the type and performance of the linkage solution, respectively.

   b. *Sea level.* There is a sea-like plane across the mountain range that can move up and down along its height (i.e., the z axis of the coordinate system) of the solutions map. The height of the sea level represents the performance of the linkage solution, i.e., the solution corresponding to the mountain part above the sea level is better than the part below it. With a rise in the sea level, the linkage solution with poor performance is submerged.

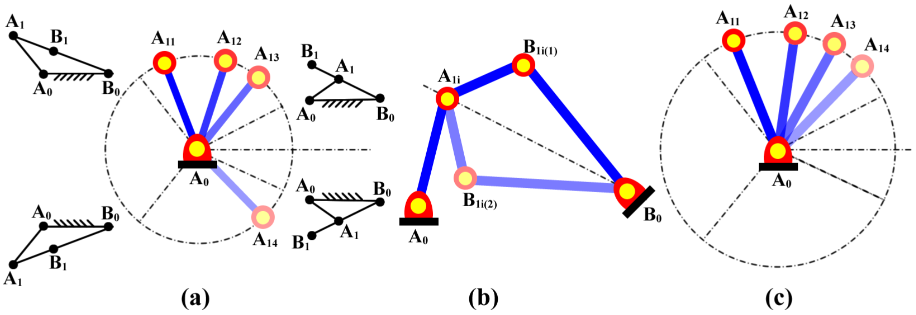


Figure 4. Linkage defects in the circuit, branch, and order. **(a)** Circuit defect. **(b)** Branch defect. **(c)** Order defect.

c. *Buoy.* The buoy is a location tool used to select a linkage solution from the solutions map. It can move freely in the X–Y plane, and its intersection with the mountain range corresponds to a linkage solution.

The method proposed here, based on the solutions map, has the characteristics of a geographical map. The concept of the *raster* and *layer* in the Geographic Information System (GIS) can thus be used to organize the data. Raster data consist of location and attribute data. As shown in Fig. 6, any grid in the solutions map is located by its row and column numbers, and corresponds to a linkage solution. As mentioned before, the row and column numbers are index numbers of points on the center curve. Based on this, the attributes of the solutions map can be divided into two types: discontinuous attributes (such as linkage and defect types) and continuous attributes (such as link length, transmission angle, acceleration, force, or their weighted values). Discontinuous data consist of a finite number of values that can be enumerated, and the values of continuous data are arbitrary within a certain range. Each discontinuous attribute is stored in the corresponding layer through the matrix, and elements of the matrix are represented by color codes corresponding to the attribute values. Continuous attributes are also stored in a layer through the matrix, and the elemental value of the matrix is the normalized performance index of the linkage. The discontinuous attributes can be represented by type map while the continuous attributes can be represented by the elevation map. The solutions map can be obtained by superimposing the two maps. Introducing the concept of the layer not only facilitates data management, but also makes the attributes of solutions map easy to query.

4. *Solution selection.* Defective solutions in the solutions map can be marked with a specific color (such as white), and solutions that yield poor performance (such as the solution with the minimum transmission angle $\gamma_{\min}$<20° at the four given positions) can be excluded by moving the sea level so that the area of the feasible solution in the solutions map can be located. Consider the position synthesis task shown in Table 4 as an example. Under the guidance of the color code of the solutions map, three groups of linkage solutions of different types can be selected by moving the buoy, as shown in Fig. 7 and Table 5.

The flowchart of the solutions map method is shown in Fig. 8. In view of the limitations of space, the generation of the coupler curve and the kinematic and dynamic analyses of the linkage are not the main contents of this paper, and are thus not discussed here.

4.2 Problems in basic solutions map

Although the solutions map can quickly locate the required types of linkage solutions and greatly improve the efficiency of synthesis, most 4R linkages synthesized based on the Burmester theory have kinematic defects, where this limits the number of available solutions. In many extreme cases, the given task of position synthesis has no solution. Consider the synthesis task given in Table 4 as an example. After excluding various defective solutions, the number of available solutions in the solutions map is 5694, accounting for only 10.86 % of the total number of solutions, 52 441, as shown in Fig. 9. Note that the solutions in the same region of the solutions map are usually highly similar, which renders options that are very limited. Therefore, it is necessary to expand the solution domain to improve the probability of obtaining feasible solutions.

5 TGA-based solutions map method

5.1 Optimization model

In the context of the problem of approximate motion synthesis, the coupler curve of the 4R linkage only needs to pass approximately through the given four positions. Similarly, although accurate positional information needs to be specified in position synthesis, it is only needed for calculation. In engineering applications, errors are usually allowed in some given positions P (x_P, y_P, δ). For example, in the de-

Table 3. Color encoding according to linkage type.

Grid color	Linkage type	legend
Sky-blue	Crank rocker	🟦
Gray	Rocker crank	🟫
Green	Double crank	🟩
Purple	Grashof double-rocker	🟪
Orange	0-0 double-rocker	🟧
Pink	0-π double-rocker	◻
Red	π-0 double-rocker	🟥
Yellow	π-π double-rocker	🟨
White	Defective or invalid linkage	□

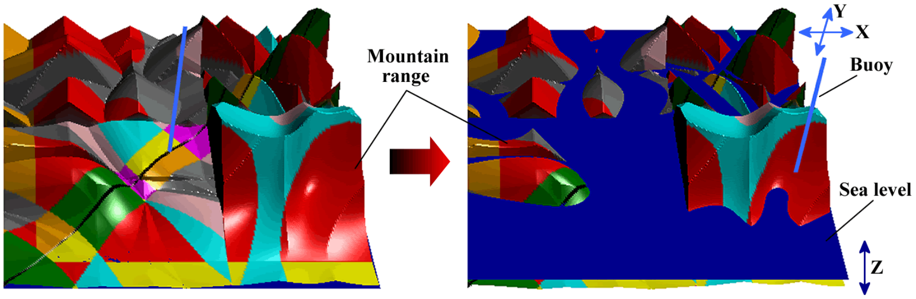


Figure 5. Solutions map for the position synthesis of the 4R linkage.

Table 4. Position synthesis task.

Position	Abscissa /m	Ordinate /m	Rotation angle /°
P_1	1.1	0	0
P_2	1.45	0.8	8
P_3	1.6	1.4	21
P_4	2.1	3.1	65

sign of the material-handling mechanism, only the accuracy of the starting and final positions of the manipulator need to be considered, and the two middle positions are only set to avoid obstacles such that errors in the middle positions are allowable. Similar examples are common, where this blurs the boundary between accurate position synthesis and approximate motion synthesis. Note that accurate position synthesis involves solving specific engineering problems. If the problem itself allows for the existence of errors, it is feasible to expand the solution domain by introducing errors, without caring about whether the problem has been transformed into that of approximate motion synthesis. For convenience of expression, the ranges of errors of x_P, y_P, and δ are denoted by $\pm\Theta x$, $\pm\Theta y$, and $\pm\Theta\delta$, respectively, as shown in Table 6.

When the positional information changes within the allowable range of error, a solutions map with different proportions of the feasible solutions can be obtained to expand the solution domain. This is a typical optimization problem, and its mathematical model can be described as follows:

$$\left\{\begin{array}{l} \boldsymbol{X}=[\lambda_1,\lambda_2,\lambda_3,\ldots,\lambda_{10},\lambda_{11},\lambda_{12}]^T \\ \text{s.t }\ \lambda_1\in\left[x_{\mathrm{p}1}-\Theta x_1, x_{\mathrm{p}1}+\Theta x_1\right], \\ \lambda_2\in\left[y_{\mathrm{p}1}-\Theta y_1, y_{\mathrm{p}1}+\Theta y_1\right], \\ \lambda_3\in[\delta_1-\Theta\delta_1,\delta_1+\Theta\delta_1]: \\ \lambda_{10}\in\left[x_{\mathrm{p}4}-\Theta x_4, x_{\mathrm{p}4}+\Theta \boldsymbol{X}_4\right], \\ \lambda_{11}\in\left[y_{\mathrm{p}4}-\Theta y_4, y_{\mathrm{p}4}+\Theta y_4\right], \\ \lambda_{12}\in[\delta_4-\Theta\delta_4,\delta_4+\Theta\delta_4] \\ \max\Gamma\left(\boldsymbol{X}\right)\frac{\Sigma\Omega^*\boldsymbol{X}}{\Sigma\Omega(\boldsymbol{X})} \end{array}\right. , \tag{22}$$

where $\boldsymbol{X}$ is the solution vector, and its elements consists of 12 position parameters λ to describe information on the four given positions, Γ is the objective function, Ω^* and Ω are the numbers of available solutions (without kinematic defects), and total solutions in the solutions map corresponding to the solution vector $\boldsymbol{X}$, respectively.

The larger the objective function Γ is, the larger is the proportion of feasible solutions in the solutions map, and the greater is the number of available solutions that can be selected. The task of optimization involves obtaining a larger objective function value of Γ by optimizing the solution vector $\boldsymbol{X}$. This study uses the genetic algorithm to expand the domain of solutions of solutions map, and its fitness function Ψ can directly use the objective function Γ, i.e., $\Psi=\Gamma$.

5.2 Genetic algorithm based on telomere operator

The genetic algorithm (Nachaoui et al., 2021; Oliveira et al., 2022) is an optimization method to simulate the mech-

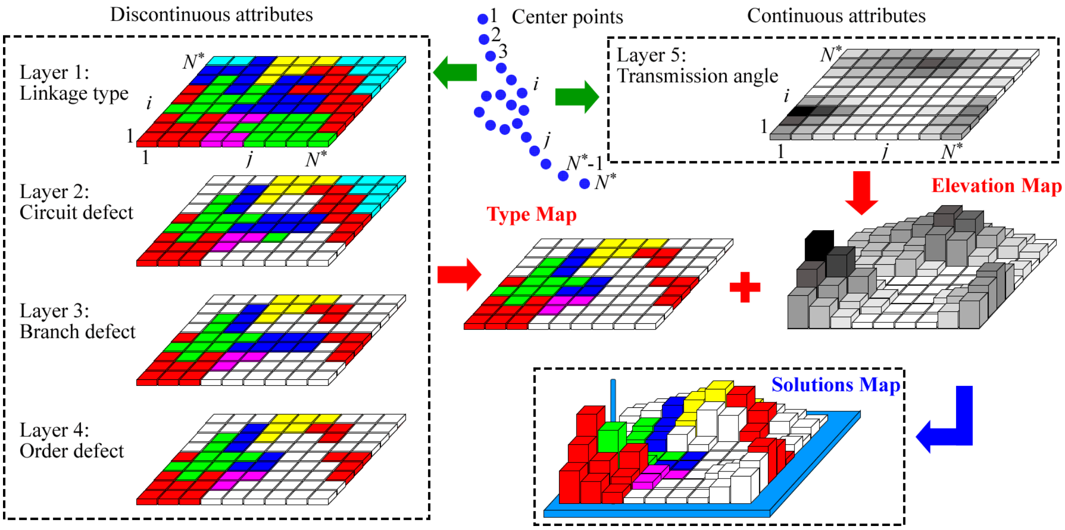


Figure 6. Model of the raster of the solutions map.

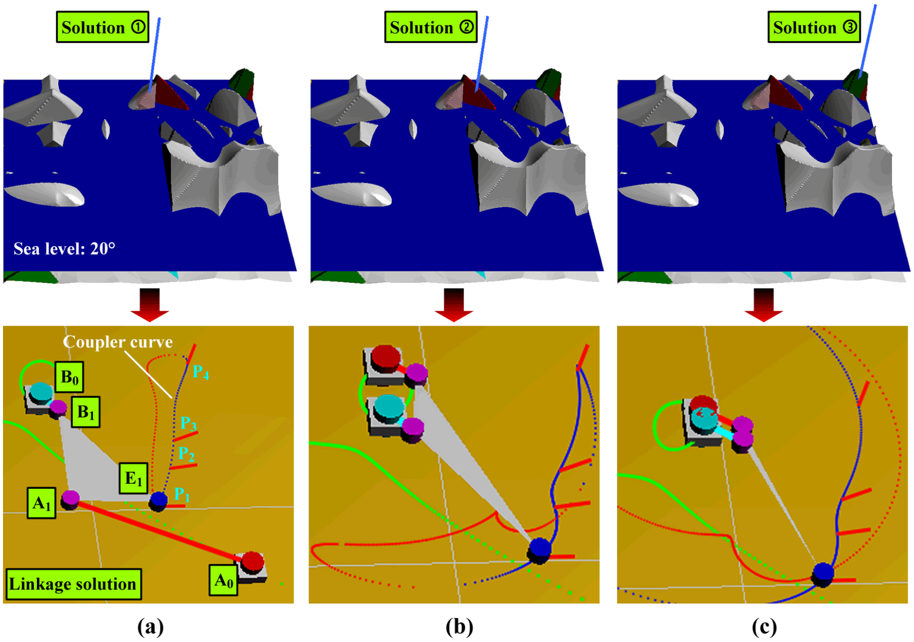


Figure 7. Linkage solutions obtained from the solutions map. **(a)** Solution 1 ($0-\pi$ double rocker). **(b)** Solution 2 ($\pi-0$ double rocker). **(c)** Solution 3 (double crank).

anism of natural evolution. The standard genetic algorithm consists of three basic operators: selection, crossover, and mutation. However, practical applications have shown that when the population is small, such an algorithm is premature. Although increasing the population can yield better results, the required population is different in different problems, and the calculation time is directly proportional to the size of the population. In previous work, catastrophe strategy has often been used to avoid the premature convergence of the genetic algorithm. This method simulates a major destruction event in natural evolution, eliminates all solutions except the optimal solution, and restarts the algorithm. Catastrophe strategy is effective for specific problems but its computational efficiency is very low. To improve the global search ability of the genetic algorithm, a new operator called the *telomere* is proposed here.

As shown in Fig. 10, telomeres are DNA-protein complexes at the end of eukaryotic chromosomes. Their function is to protect the chromosomal structure and control the cell-division cycle. Every time a cell divides, the length of telomeres decreases. When telomeres are exhausted, the cells will gradually stop dividing due to the destruction of the DNA

Table 5. Coordinate-related information of the linkage solutions.

Solution ID	Point coordinates (Abscissa/m, Ordinate/m)				Linkage type	γ_{min}/deg
	A_0	A_1	B_0	B_1		
1	(2.56047, −1.33283)	(−0.42227, 0.06935)	(−0.67032, 2.31346)	(−0.41089, 1.98080)	$0-\pi$ double rocker	38.29059
2	(−0.59232, 3.15632)	(−0.20345, 2.88538)	(−0.64224, 2.32030)	(−0.36253, 1.98212)	$\pi-0$ double rocker	33.31090
3	(−0.25936, 2.90638)	(0.26972, 2.57717)	(−0.27455, 2.68865)	(0.24369, 2.33214)	Double crank	37.30242

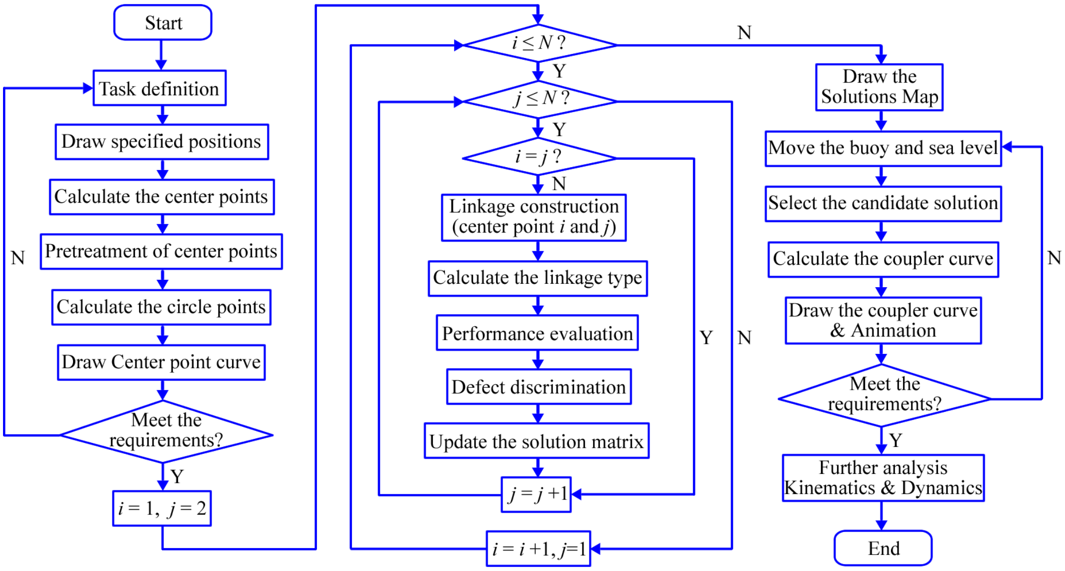

Start
Task definition
Draw specified positions
Calculate the center points
Pretreatment of center points
Calculate the circle points
Draw Center point curve
Meet the requirements?
N
Y
$i=1,\ j=2$
$i \le N$?
Y
$j \le N$?
Y
$i=j$?
N
Linkage construction (center point i and j)
Calculate the linkage type
Performance evaluation
Defect discrimination
Update the solution matrix
$j=j+1$
$i=i+1, j=1$
Y
N
N
Draw the Solutions Map
Move the buoy and sea level
Select the candidate solution
Calculate the coupler curve
Draw the coupler curve & Animation
Meet the requirements?
N
Y
Further analysis Kinematics & Dynamics
End

Figure 8. Flowchart of the solutions map method.

Table 6. Allowable errors in the position synthesis task.

Position	Abscissa/m	Ordinate m	Rotation angle/deg
P_1	$x_{P_1} \pm \Theta x_1$	$y_{P_1} \pm \Theta y_1$	$\delta_1 \pm \Theta \delta_1$
P_2	$x_{P_2} \pm \Theta x_2$	$y_{P_2} \pm \Theta y_2$	$\delta_2 \pm \Theta \delta_2$
P_3	$x_{P_3} \pm \Theta x_3$	$y_{P_3} \pm \Theta y_3$	$\delta_3 \pm \Theta \delta_3$
P_4	$x_{P_4} \pm \Theta x_4$	$y_{P_4} \pm \Theta y_4$	$\delta_4 \pm \Theta \delta_4$

structure. Therefore, the telomere is also known as the mitotic clock. In the standard genetic algorithm, not all individuals participate in the crossover and mutation operations. Their participation depends on the probabilities of crossover and mutation. Some old individuals thus have no chance of producing new individuals in successive generations of evolution. The core idea of the *telomere operator* is to eliminate these old and useless individuals, and replace them with new ones. Its basic principles are as follows:

1. A variable κ, describing the "telomere length", is added at the end of each solution vector, that is, $X = [\lambda_1, \lambda_2, \ldots, \lambda_{12}, \kappa]^T$.

2. The initial length of the telomere is M, and its length κ decreases by 1 in each evolutionary generation. When its length is less than zero, the individual is eliminated and replaced with a randomly generated individual.

3. If the crossover or mutation operation can produce a new individual that is different from the parent, the length of the telomere of the new individual is reset to M.

In applications, the performance of the telomere operator can be further improved by the following methods:

1. The initial length of the telomere is variable. If the fitness of the optimal individual in the current generation does not change compared with that in the previous generation, $M = M - 1$; if M is less than zero, $M = 0$. Note that because the condition for individual elimination is $\kappa < 0$ (not $\kappa = 0$), $M = 0$ does not mean that all individuals are regenerated. This method assumes that the length of the telomere is affected by environmental stress. When evolution slows down, the algorithm strengthens individual extinction and regeneration caused by the telomere.

2. When the old individual is eliminated, a new individual is produced by it through non-uniform mutation (Chauhan et al., 2021; Ma, 2021). This method has a

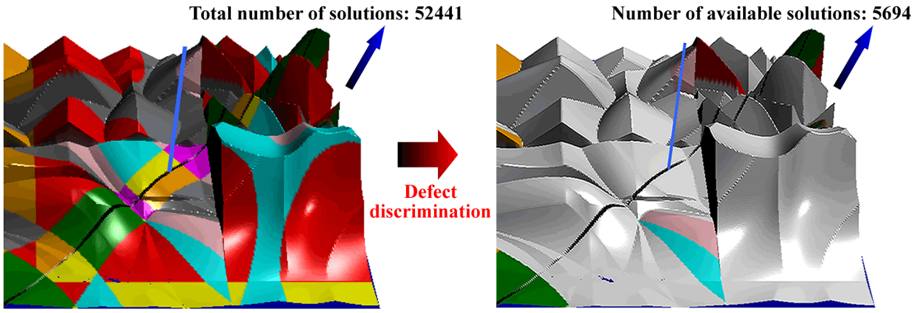

Total number of solutions: 52441
Defect discrimination
Number of available solutions: 5694

Figure 9. Distribution of solutions without kinematic defects in the solutions map.

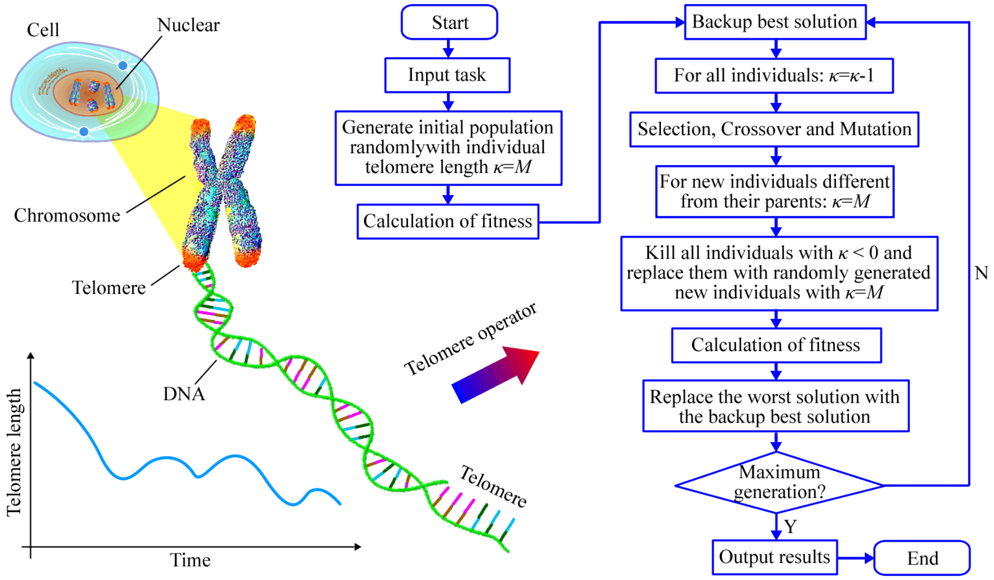

Cell
Nuclear
Chromosome
Telomere
DNA
Telomere length
Time
Telomere operator
Telomere
Start
Input task
Generate initial population randomlywith individual telomere length κ=M
Calculation of fitness
Backup best solution
For all individuals: κ=κ-1
Selection, Crossover and Mutation
For new individuals different from their parents: κ=M
Kill all individuals with κ < 0 and replace them with randomly generated new individuals with κ=M
Calculation of fitness
Replace the worst solution with the backup best solution
Maximum generation?
N
Y
Output results
End

Figure 10. Biological structure and algorithmic simulation of the telomere.

strong capability of global search in the early stage, and changes to local fine search in the later stage. The position of mutation s ($s = 1, 2, \ldots, 12$) of the old individual is randomly selected, and its element λ_s is updated using the following equation:

$$\lambda_s = \begin{cases} \lambda_s + \Delta_k(g,\ U_{max} - \lambda_s), & \text{if } r_{1k} \le 0.5 \\ \lambda_s - \Delta_k(g,\ \lambda_s - U_{min}) & \text{if } r_{1k} > 0.5 \end{cases}, \qquad (23)$$

and

$$\Delta_k(g,\ \chi) = \chi \cdot \left(1 - r_{2k}^{\left(1-\frac{g}{G}\right)^b}\right), \qquad (24)$$

where U_{max} and U_{min} are the upper and lower limits of λ_s, respectively, r_{1_k} and r_{2_k} are independent and identically distributed random variables that obey a uniform distribution in the interval $[0, 1]$, and g and G are the current generation and total number of generations of the genetic algorithm, respectively.

Based on the above algorithm, we take $\Theta x_i = 0.1$, $\Theta y_i = 0.1$, and $\Theta\delta_i = 5°$ $(i = 1-4)$ to optimize the solutions map shown in Fig. 9, and the result is shown in Fig. 11. It shows that the proportion of available solutions after optimization increases from 0.108579 to 0.871961.

To further verify the effectiveness of the telomere operator, the results of optimization of the standard genetic algorithm (GA) and the telomere genetic algorithm (TGA) were compared with populations of $P_s = 20$ and $P_s = 40$, respectively, as shown in Table 7. Each group of optimization tests was repeated 20 times, and the parameters used in the calculation were as follows: total number of generations $G_t = 30$, crossover probability $P_c = 0.6$, mutation probability $P_m = 0.1$, and initial telomere length $M = 3$. In addition, float encoding, traditional roulette wheel selection, arithmetical crossover, a non-uniform mutation operator, and the elitist strategy were used by both the GA and the TGA during calculations. To compare their performance, the average ζ_{av} was used to measure the overall fitness values obtained by the algorithm while the average deviation ζ_{ad} was used to express

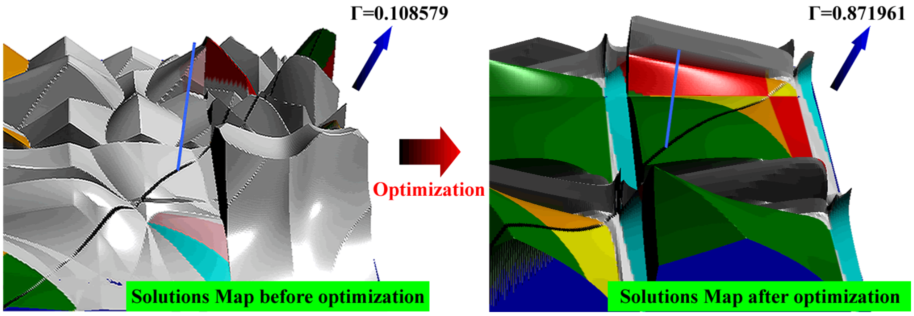

Γ=0.108579
Γ=0.871961
Optimization
Solutions Map before optimization
Solutions Map after optimization

Figure 11. Genetic algorithm-based optimization of solutions map based on the telomere operator.

fluctuations in these values:

$$\zeta_{\mathrm{av}} = \frac{\sum_{i=1}^{20} y_i}{20}, \tag{25}$$

$$\zeta_{\mathrm{ad}} = \frac{\sum_{i=1}^{20} |y_i - \zeta_{\mathrm{av}}|}{20}, \tag{26}$$

where y_i is the best fitness of the algorithm in the ith test. Table 7 shows that when the population was 40, the performance of the TGA was equivalent to that of the GA. The average fitness and average deviation values of both were close. However, when the population was 20, the GA struggled to converge to the global optimal solution and there were large fluctuations in the data, while the TGA could still calculate normally. Its average fitness was 0.08449 higher than that of the GA and the average deviation in it was only 17.8 % of that in the GA. The telomere operator can thus avoid the premature convergence of the GA, especially when the population is small. This means that the TGA can use a small population to obtain approximately optimal results to save a significant amount of computing time.

The typical fitness curves of the GA and TGA are shown in Fig. 12. Under normal circumstances, the optimal fitness value of the population rapidly improves in the initial stage of calculation of the genetic algorithm. Consider the curves of convergence "GA (1)" of the standard genetic algorithm and "TGA (1)" of the telomere genetic algorithm as examples: the initial fitness values of GA (1) and TGA (1) were below 0.2, but both increased to more than 0.7 within the first three generations. However, another form of the curve of convergence is also very common, where the GA prematurely falls into the local optimal solution. Consider the curves of convergence "GA (2)" of the GA and "TGA (2)" of the TGA as examples: when the GA falls into the local optimal solution, all individuals in the population are highly similar and the crossover operation can no longer produce a new solution. Although GA (1) jumped out of the local optimal solution through the mutation operation in the 20th generation, it fell into another local optimal solution and finally failed to achieve global convergence. By sharp contrast, TGA (2) could still generate new solutions with the help of the telomere operator after falling into the local optimal solution, jumping out of the local optimal solution many times until it achieved global convergence. We can glean important information from TGA (2): a large number of local optimal solutions may be distributed in the solution domain of optimization problems of the solutions map, where this is similar to the domain of solutions of multi-modal functions. Based on this assumption, we think that increasing the population size helps to expand the search scope of the algorithm. However, the telomere operator, which can enable the algorithm to jump repeatedly between local optimal solutions, can achieve a similar effect and is clearly more efficient.

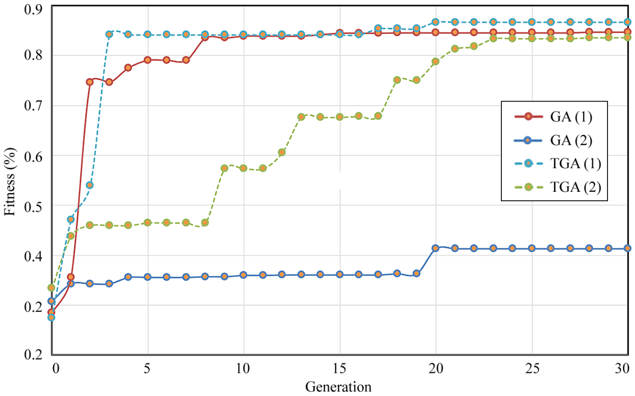


Figure 12. Typical convergence curves of the GA and TGA.

5.3 Obtaining solutions maps with more options

In general, the GA can only obtain a solution with the highest fitness after convergence. However, a solutions map with a high fitness does not necessarily mean that a satisfactory solution can be obtained. In the problem of synthesis of linkage positions, the evaluation of the solution is multi-faceted. In addition to the absence of kinematic defects, the linkage solution needs to satisfy geometric constraints, such as the length of the linkage and position of installation, as well as the requirements of mechanical performances, such as kinematics

Table 7. Comparison of results of optimization of the GA and TGA.

Test ID	GA		TGA	
	Population size $P_\mathrm{s} = 20$	Population size $P_\mathrm{s} = 40$	Population size $P_\mathrm{s} = 20$	Population size $P_\mathrm{s} = 40$
1	0.378284087	0.887477551	0.859090909	0.8671
2	0.817448388	0.855398792	0.861760409	0.845305991
3	0.613193772	0.869659864	0.855567713	0.845143802
4	0.861171983	0.839647487	0.843101211	0.855937872
5	0.821052145	0.858483205	0.796739304	0.851473923
6	0.853298844	0.841573854	0.855367901	0.848620113
7	0.840459184	0.850110973	0.874886779	0.849397852
8	0.847413578	0.84416583	0.786193848	0.870360735
9	0.773501544	0.854957248	0.874674882	0.859487866
10	0.784233056	0.808170066	0.856443491	0.829259711
11	0.842877782	0.845972703	0.849732774	0.88055293
12	0.833070234	0.86542606	0.869961169	0.846781928
13	0.817854605	0.86427258	0.878960503	0.872874434
14	0.843851564	0.789867145	0.815854592	0.858446046
15	0.843543015	0.842710005	0.849264198	0.855105105
16	0.879003463	0.871948289	0.844423079	0.849507736
17	0.532483775	0.849771901	0.880223457	0.862379808
18	0.553477321	0.835394965	0.849298469	0.85831176
19	0.870815177	0.854970504	0.858034393	0.873742496
20	0.679760639	0.885496831	0.817084298	0.865140604
ζ_av	0.764339708	0.850773793	0.848833169	0.857246536
ζ_ad	0.106449895	0.0160353	0.018960268	0.009593132

and dynamics. Therefore, even if the proportion of available solutions of the optimized solutions map is very high, the solutions it provides may still be completely rejected in the subsequent analysis due to limitations imposed by other conditions of evaluation. A more feasible method is to generate multiple solutions maps to provide designers with more candidate solutions. If the optimal solutions map cannot provide a satisfactory solution, we can continue to search for feasible solutions in other solutions maps. *Niche* technology is used in this study to obtain multiple solutions maps through a single optimization. This can help to maintain the diversity of the population in the GA to prevent all individuals from converging to the same solution. Current niche technologies include the sharing function and the crowding mechanism (Che et al., 2021; Liu et al., 2020). This study uses the sharing function method. The sharing function Λ is used to measure the similarity between solutions:

$$\Lambda\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right) = \begin{cases} 1 - \frac{\Pi_1(\boldsymbol{X}_i,\boldsymbol{X}_j)}{{\in}_1} & \text{if } \Pi_1 < {\in}_1, \Pi_2 \geq {\in}_2 \\ 1 - \frac{\Pi_2(\boldsymbol{X}_i,\boldsymbol{X}_j)}{{\in}_2} & \text{if } \Pi_1 \geq {\in}_1, \Pi_2 < {\in}_2 \\ 1 - \frac{\Pi_1(\boldsymbol{X}_i,\boldsymbol{X}_j)\Pi_2(\boldsymbol{X}_i,\boldsymbol{X}_j)}{{\in}_1{\in}_2} & \text{if } \Pi_1 < {\in}_1, \Pi_2 < {\in}_2 \\ 0 \quad \text{else} \end{cases}, \tag{27}$$

where Π_1 represents the genotypic distance between solution vectors $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, Π_2 represents the phenotypic distance between them, and μ_1 and μ_2 are the maximum individual distances of the genotype and the phenotype in the niche technology, respectively.

The modified fitness function Ψ^* can then be expressed as:

$$\Psi^*(\boldsymbol{X}_i) = \frac{\Psi(\boldsymbol{X}_i)}{\sum_{j=1}^{P_\mathrm{s}} \Lambda\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right)}, \tag{28}$$

when the value of the sharing function between an individual and all other individuals is large, its fitness is reduced to prevent the individual with the highest fitness from assimilating the entire population and maintain the diversity of the population. Therefore, niche technology can be used to obtain the optimal solution as well as a series of local optimal solutions that are different from one another.

In Eq. (27), the measurement of similarity by the sharing function depends on two distance functions, Π_1 and Π_2. Genotypic similarity between individuals is represented by Π_1. For a real-coded solution vector, Π_1 can be expressed by the Euclidean distance:

$$\Pi_1(\boldsymbol{X}_1, \boldsymbol{X}_2) = \sqrt{(\boldsymbol{X}_1 - \boldsymbol{X}_2)(\boldsymbol{X}_1 - \boldsymbol{X}_2)^T}. \tag{29}$$

In this study, Π_1 is not necessary because we attend more to the similarity of phenotypes than genotypes. By contrast, Π_2 is important for representing (phenotypic) similarity. In the traditional sharing function method, Π_2 is usually expressed by a fitness distance. However, the fitness function

Ψ here represents the proportion of available solutions in the solutions map, and not the similarity between solutions maps. Therefore, a pseudo-histogram method for measuring the similarity between solutions maps is proposed. The histogram is used to describe the frequency distribution of different colors in an image. The solutions map has the characteristics of an image, and as a result, the histogram method can be used to describe the number distribution of eight kinds of linkages in the solutions map. However, the solutions map is not an image in the real sense. Hence, the corresponding histogram is called a "pseudo-histogram" in this study. If the pseudo-histogram is used as the feature of the solutions map, the similarity between any two maps can be measured by the Bhattacharyya coefficient:

$$\Pi_2(\boldsymbol{X}_1, \boldsymbol{X}_2) = \sum\nolimits_{i=1}^{8} \sqrt{\Xi(\boldsymbol{X}_1)_i \times \Xi(\boldsymbol{X}_2)_i}, \qquad (30)$$

where $\Xi\ (\boldsymbol{X})$ represents the histogram vector corresponding to the solution vector $\boldsymbol{X}$, which has eight elements. Each represents the number of solutions with a specific linkage type. The subscript i of Ξ indicates the index of the ith element in the histogram vector.

By embedding the sharing function based on the pseudo-histogram into the fitness function of the TGA, multiple groups of solutions maps can be obtained in a single calculation. Consider once again the position synthesis task in Table 4 as an example. Four groups of typical solutions maps obtained based on the above algorithm are shown in Fig. 13. It is clear that the Bhattacharyya coefficient calculated from the pseudo-histogram can adequately reflect the similarity between different solutions maps. Moreover, the niche realized by the pseudo-histogram can avoid the convergence of the GA to the same solution.

To further verify the influence of the niche technology based on the pseudo-histogram on the GA, the results of optimization of the GA and the TGA were compared with populations of $P_\mathrm{s} = 20$ and $P_\mathrm{s} = 40$, as shown in Table 8. It shows that the niche technology based on the sharing function method had a significant impact on the GA. The maximum fitness value obtained by optimization was generally low and unstable. Even if the population was expanded to 40, the fitness value obtained from three tests (ID = 2, 4, 9) was below 0.7 because the sharing function method changed the standard of fitness evaluation. The fitness of a solution depends not only on the objective function, but also on the similarity between it and the entire population. According to the previous assumptions, many local optimal solutions might have been distributed in the solution domain of optimization problems of the solutions map, and the use of the sharing function generally reduces the fitness of these solutions. Therefore, only when the algorithm converges to the real global optimal solution, can we obtain stable and high-quality results of optimization. The standard GA can usually converge only to local optimal solutions with different fitness values after each optimization, where the fitness values of these solutions are generally low and different. This causes the results of optimization to greatly fluctuate. Although the expansion of the population helps to expand the search scope of the algorithm, it is difficult to ensure that part of the randomly generated individuals in the initial population fall into the vicinity of the global optimal solution for further search. In this case, the algorithm is still trapped in widely distributed local optimal solutions with low fitness. By contrast, the telomere operator can enable the GA to jump between local optimal solutions and thus can always converge to the global optimal solution. This renders the algorithm stable. For these reasons, the telomere operator is insensitive to population size, and this indicates that the required solutions maps can still be obtained with a smaller population to reduce computation time.

5.4 *BurLink* software based on solutions map method

The solutions map is a computer-aided method of synthesis. To improve the efficiency of synthesis, the linkage synthesis software *BurLink* was developed based on the solutions map, as shown in Fig. 14. It offers two views. The right view is the solutions map and the left view shows the linkage solution selected from it. The movement of the buoy and the sea level in the solutions map can be controlled by sliders. When the buoy moves, the linkage solution and the coupler curve are updated immediately. The software can output the coordinates of each hinge on the linkage and show the corresponding animation.

In addition to the calculation of points on the center and the circle, each linkage solution in the solutions map requires the calculation and visualization of the types of linkage and defect, and transmission-related performance. If the time needed for the calculation is too long, the synthesis efficiency of the software decreases. Most of the computation time is spent on the visualization of the solutions map. To shorten this, the Java3D API is used to generate the solutions map. Consider the solutions map shown in Fig. 5 as an example. *BurLink* can complete all calculations and the 3D display of 52 441 solutions within 300 ms (test conditions: 11th Gen Intel(R) Core(TM) i7-1165G7 at 2.80 GHz; RAM 16 GB). It also supports the TGA in expanding the domain of solution of the solutions map. After TGA optimization, users can extract information on any individual in the population and update the map.

5.5 Discussion

According to the analysis in Sect. 1, past methods can be divided into three categories: traditional method based on region-based segmentation technology, type map based on solution domain analysis, and approximate motion synthesis based on optimization. Both the type map and TGA-based solutions map can be used to analyze the solution domain; thus their synthesis efficiencies are much higher than that

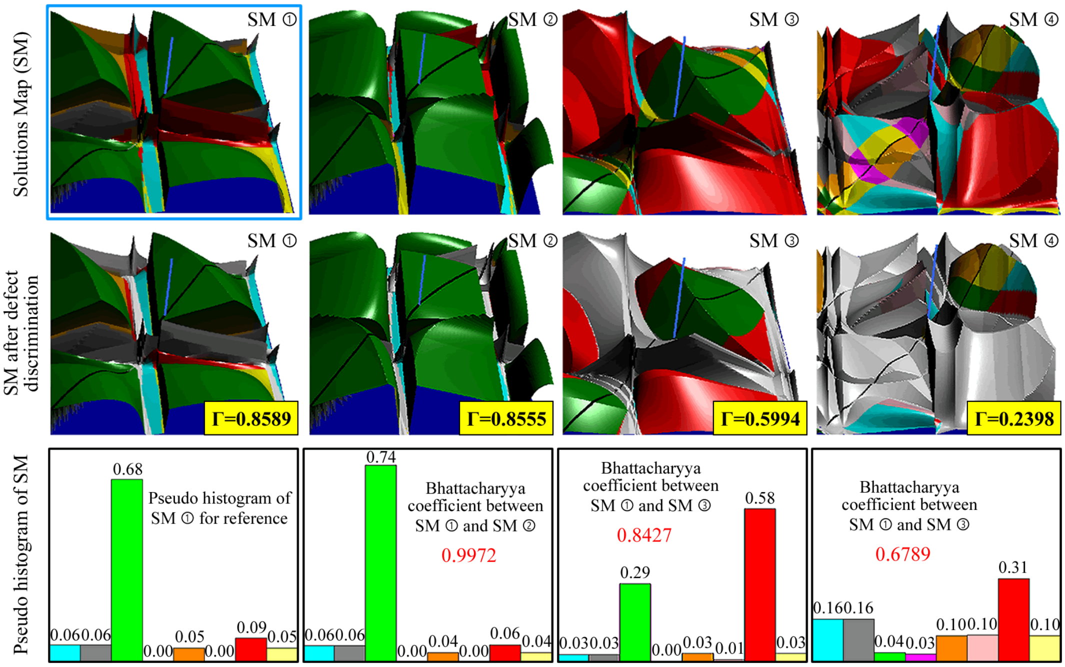
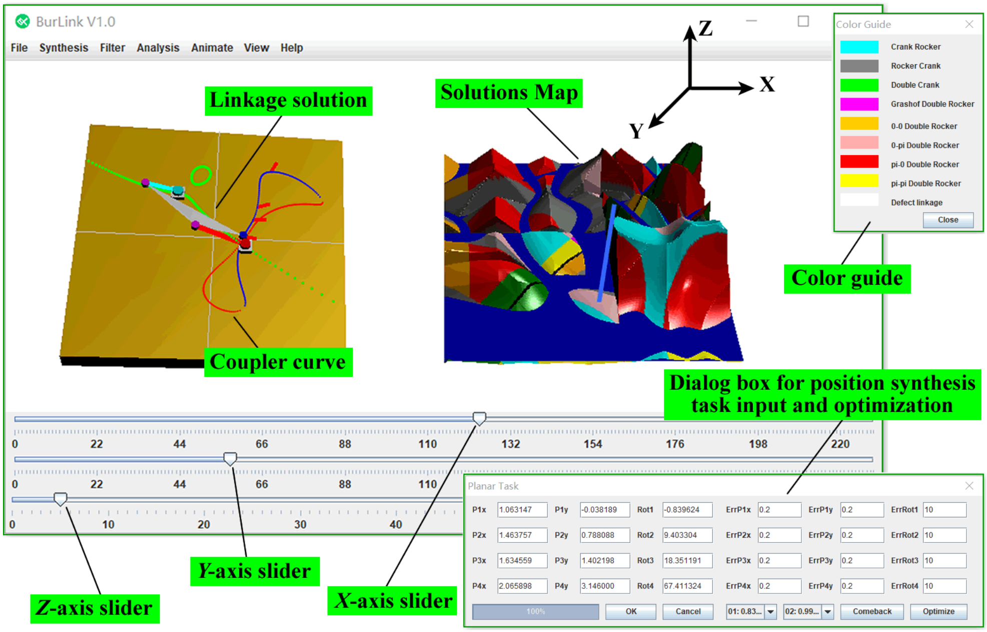


Figure 13. Solutions maps generated by the niche genetic algorithm based on the pseudo-histogram method.

Table 8. Comparison of results of niche optimization of the GA and TGA.

Test ID	GA		TGA	
	Population size $P_\mathrm{s} = 20$	Population size $P_\mathrm{s} = 40$	Population size $P_\mathrm{s} = 20$	Population size $P_\mathrm{s} = 40$
1	0.808140496	0.719515621	0.864022039	0.854201389
2	0.845178814	0.688117222	0.871671598	0.848327124
3	0.847435393	0.808539516	0.805541023	0.845135764
4	0.819129855	0.613161143	0.791941971	0.834410095
5	0.72313888	0.79294896	0.880453601	0.76589113
6	0.378567574	0.872318339	0.814721197	0.865165839
7	0.783488051	0.760982924	0.878007221	0.794923907
8	0.617957746	0.822096614	0.797713479	0.811727568
9	0.65781804	0.678953044	0.835121086	0.775965734
10	0.83415333	0.806904164	0.8374514	0.813102075
11	0.762416295	0.786905787	0.820455668	0.863822243
12	0.833188657	0.826113212	0.830851801	0.860283435
13	0.742630385	0.85883183	0.850974277	0.794999717
14	0.775725	0.789474375	0.82898474	0.869401192
15	0.847920605	0.845536032	0.82237643	0.874622222
16	0.802203709	0.87267562	0.813134105	0.841278983
17	0.841179666	0.832508151	0.832767857	0.840127081
18	0.749836018	0.799855215	0.84064	0.765636095
19	0.783424238	0.835722012	0.841445923	0.876876502
20	0.795826689	0.798990032	0.84377729	0.853818085
ζ_av	0.762467972	0.790507491	0.835102635	0.832485809
ζ_ad	0.070491087	0.049644232	0.019253808	0.030515444



P1x	1.063147	P1y	-0.038189	Rot1	-0.839624	ErrP1x	0.2	ErrP1y	0.2	ErrRot1	10
P2x	1.463757	P2y	0.788088	Rot2	9.403304	ErrP2x	0.2	ErrP2y	0.2	ErrRot2	10
P3x	1.634559	P3y	1.402198	Rot3	18.351191	ErrP3x	0.2	ErrP3y	0.2	ErrRot3	10
P4x	2.065898	P4y	3.146000	Rot4	67.411324	ErrP4x	0.2	ErrP4y	0.2	ErrRot4	10

Figure 14. Software used for the synthesis of the linkage position.

of the traditional, blind method. Compared with the type map, the TGA-based solutions map adds a dimension to visualize continuous attributes of the linkage, and can expand the solution domain through the TGA and niche technology. Although the TGA-based solutions map also uses the optimization algorithm, its objective of optimization is the solution domain and not a single linkage solution. In addition, it blurs the boundary between accurate position synthesis and approximate motion synthesis, which means that the two problems can be solved by a unified method in some cases. When the allowable errors in all guidance positions are zero, the TGA-based solutions map solves the problem of accurate position synthesis; when errors are introduced to part of the guidance positions, it solves the mixed problem of exact–approximate synthesis; when errors are introduced to all guidance positions, it solves the problem of approximate motion synthesis.

Compared with previous methods, the highlight of the proposed TGA-based solutions map is that it can provide more candidate solutions for the problem of the four-position synthesis of the 4R linkage to satisfy the needs of subsequent design. Consider the synthesis task in Table 4 as an example once again. Statistical results show that the number of candidate solutions that can be obtained by the type map or the basic solutions map is 5694, whereas the number of candidate solutions that can be obtained by the TGA-based solutions map is 25 670. Considering that solutions in the area of the same linkage type of the type map or the solutions map are highly similar, it is not rigorous enough to compare the two methods in terms of the number of candidate solutions only. If the linkage needs to be free of motion-related defects and the minimum transmission angle at the four specified positions is greater than 30°, the region in the solution domain that satisfies the above constraints and has the same linkage type can be called the candidate region. As is shown in Fig. 15, the type map can provide 9 candidate regions for the synthesis task in Table 6 while the solutions map based on the TGA and niche technology (population size, $P_s = 20$) can provide 20 solution domains and a total of 208 candidate regions. In particular, the solutions map can find solutions of all linkage types, but the type map cannot find the rocker crank and Grashof double-rocker linkage solutions that satisfy the given requirements. Note that only the transmission angle is limited here. If the ratio of the link length, velocity and acceleration of the guidance point, force of the driving link, and other attributes are limited, the nine candidate regions provided by the type map are likely to be significantly reduced or may even lead to no solution. By contrast, after the expansion of the solution domain, the solutions map still has a larger selection space owing to the large number of regions of candidate solutions.

6 Conclusions

In this paper, the authors examined the computer-aided synthesis of the planar 4R linkage. The main conclusions can be summarized as follows:

1. A solutions map method was proposed. It can reveal the distribution of the linkage solutions of different attributes, including discontinuous attributes (such as linkage and defect types) and continuous attributes (such as link length, transmission angle, acceleration,

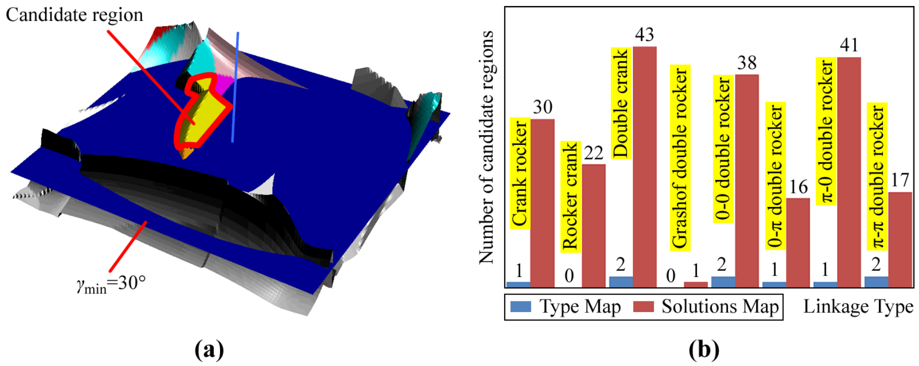


Figure 15. Analysis of candidate regions. (**a**) Schematic diagram of candidate regions in the solutions map. (**b**) Number of candidate regions with different types of linkages.

and force, or their weighted values) so that the required linkage solutions can be quickly located.

2. A method of generating the solutions map was proposed. The center curve was first calculated by projective geometry and the circle curve was then obtained by vector elimination. A defect discrimination algorithm was proposed based on this which can quickly eliminate defective linkage solutions from the solutions map to improve the efficiency of linkage synthesis.

3. An improved genetic algorithm (GA) based on the telomere operator was proposed and used to expand the domain of solution of the solutions map. A niche construction method based on a pseudo-histogram was proposed based on this, such that more solutions maps and candidate solutions can be obtained after optimization. The results showed that the telomere genetic algorithm (TGA) significantly outperformed the traditional GA regarding the problem of expanding the domain of solutions of the solutions map.

4. The TGA-based solutions map was used to develop the software *BurLink* for linkage synthesis.

5. Compared with previous research, the proposed TGA-based solutions map can provide more candidate solutions for the problem of the four-position synthesis of the 4R linkage. Moreover, it blurs the boundary between accurate position synthesis and approximate motion synthesis, which means that the two problems can be solved by a unified method in some cases.

6. The TGA-based solutions map can also be applied to the finite positional synthesis of any linkage with at least two free variables, including planar and spatial linkages. The specific method of operation is consistent with that in this study. It is first necessary to specify two free variables in the linkage synthesis model to form the solutions map and to then use the TGA to optimize it.

Code and data availability. All the data used in this paper can be obtained by request from the corresponding author.

Author contributions. YZ, LX, and GW wrote the paper and participated in the algorithm discussion of constructing the telomere genetic algorithm (TGA). YZ wrote the fast validation program of TGA under MATLAB. GW proposed the concept of TGA and solutions map, developed *BurLink* software and transplanted YZ's TGA program from MATLAB to *BurLink*. LX provided all the test data required for the paper based on *BurLink*. FZ, YS, and HZ edited and verified all the formulas and pictures used in this paper.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgements. This research has been supported by the Shandong Provincial Natural Science Foundation (grant no. ZR2020QE163), the Program of Shandong Provincial Key Laboratory of Horticultural Machineries and Equipment (grant no. YYJX-2019-01), the Shandong Provincial Key Research and Development Program (grant no. 2018GNC112008) and the China Agriculture Research System of MOF and MARA (grant no. CARS-27).

Financial support. This research has been supported by the Shandong Provincial Natural Science Foundation (grant no. ZR2020QE163), the Program of Shandong Provincial Key Laboratory of Horticultural Machineries and Equipment (grant no. YYJX-2019-01), the Shandong Provincial Key Research and Development Program (grant no. 2018GNC112008), and the China Agriculture Research System of MOF and MARA (grant no. CARS-27).

Review statement. This paper was edited by Francisco Romero and reviewed by two anonymous referees.

References

Baskar, A. and Bandyopadhyay, S.: A homotopy-based method for the synthesis of defect-free mechanisms satisfying secondary design considerations, Mech. Mach. Theory, 133, 395–416, https://doi.org/10.1016/j.mechmachtheory.2018.12.002, 2019.

Cao, Y. and Han, J.: Solution region-based synthesis methodology for spatial HCCC linkages, Mech. Mach. Theory, 143, 103619, https://doi.org/10.1016/j.mechmachtheory.2019.103619, 2020.

Cera, M. and Pennestrì, E.: Generalized Burmester points computation by means of Bottema's instantaneous invariants and intrinsic geometry, Mech. Mach. Theory, 129, 316–335, https://doi.org/10.1016/j.mechmachtheory.2018.07.011, 2018.

Cera, M. and Pennestrì, E.: The mechanical generation of planar curves by means of point trajectories, line and circle envelopes: A unified treatment of the classic and generalized Burmester problem, Mech. Mach. Theory, 142, 103580, https://doi.org/10.1016/j.mechmachtheory.2019.103580, 2019.

Chauhan, S., Singh, M., and Aggarwal, A. K.: Diversity driven multi-parent evolutionary algorithm with adaptive non-uniform mutation, J. Exp. Theor. Artif. In., 33, 775–806, https://doi.org/10.1080/0952813X.2020.1785020, 2021.

Che, Z. H., Chiang, T. A., and Lin, T. T.: A multi-objective genetic algorithm for assembly planning and supplier selection with capacity constraints, Appl. Soft Comput., 101, 107030, https://doi.org/10.1016/j.asoc.2020.107030, 2021.

Erdman, A. G. and Loftness, P. E.: Synthesis of linkages for cataract surgery: storage, folding, and delivery of replacement intraocular lenses (IOLs), Mech. Mach. Theory, 40, 337–351, https://doi.org/10.1016/j.mechmachtheory.2004.07.006, 2005.

Filemon, E.: Useful ranges of centerpoint curves for design of crank-and-rocker linkages, Mech. Mach. Theory, 7, 47–53, https://doi.org/10.1016/0094-114X(72)90015-8, 1972.

Ge, Q. J., Purwar, A., Zhao, P., and Deshpande, S.: A Task-driven approach to unified synthesis of planar four-bar linkages using algebraic fitting of a pencil of G-manifolds, J. Comput. Inf. Sci. Eng., 17, 031011, https://doi.org/10.1115/1.4035528, 2017.

Gogate, G. R. and Matekar, S. B.: Optimum synthesis of motion generating four-bar mechanisms using alternate error functions, Mech. Mach. Theory, 54, 41–61, https://doi.org/10.1016/j.mechmachtheory.2012.03.007, 2012.

Gogate, G. R. and Matekar, S. B.: Unified synthesis of Watt-I six-link mechanisms using evolutionary optimization, J. Mech. Sci. Technol., 28, 3075–3086, https://doi.org/10.1007/s12206-014-0715-0, 2014.

Gupta, K. C. and Beloiu, A. S.: Branch and circuit defect elimination in spherical four-bar linkages, Mech. Mach. Theory, 33, 491–504, https://doi.org/10.1016/S0094-114X(97)00078-5, 1998.

Han, J. and Cao, Y.: Solution Region Synthesis Methodology of RCCC Linkages for Four Poses, Mech. Sci., 9, 297–305, https://doi.org/10.5194/ms-9-297-2018, 2018.

Jia, G., Huang, H., Wang, S., and Li, B.: Type synthesis of plane-symmetric deployable grasping parallel mechanisms using constraint force parallelogram law, Mech. Mach. Theory, 161, 104330, https://doi.org/10.1016/j.mechmachtheory.2021.104330, 2021.

Jin, Z. and Shi, Y.: Computer drafting of center curve, Journal of North China University of Technology, 3, 76–85, 1991.

Liang, C. and Chen, H.: Calculation and Design of Planar Linkage, Higher Education Press, Beijing, China, ISBN: 7-04-003936-2, 1993.

Liu, H. and Han, J.: Solution region synthesis methodology of spatial 1CS-4SS linkages for six given positions, Mech. Mach. Theory, 162, 104369, https://doi.org/10.1016/j.mechmachtheory.2021.104369, 2021.

Liu, J., Liu, J., Yan, X., and Peng, B.: A heuristic algorithm combining Pareto optimization and niche technology for multi-objective unequal area facility layout problem, Eng. Appl. Artif. Intel., 89, 103453, https://doi.org/10.1016/j.engappai.2019.103453, 2020.

Martin, D. T. and Murray, A. P.: Developing classifications for synthesizing, refining, and animating planar mechanisms, in: ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 29 September–2 October 2002, Montreal, Quebec, Canada, 1397–1403, https://doi.org/10.1115/DETC2002/MECH-34372, 2002.

Ma, Z., Li, S., Wang, Y., and Yang, Z.: Component-level construction schedule optimization for hybrid concrete structures, Automat. Const., 125, 103607, https://doi.org/10.1016/j.autcon.2021.103607, 2021.

McCarthy, J. M.: Kinematic Synthesis, in: 21st Century Kinematics, edited by: Antman, S. S., Marsden, J. E., and Sirovich, L., Springer, London, e-ISBN: 978-1-4419-7892-9, ISBN: 978-1-4419-7891-2, https://doi.org/10.1007/978-1-4419-7892-9, 2013.

Nachaoui, M., Afraites, L., and Laghrib, A.: A regularization by denoising super-resolution method based on genetic algorithms, Signal Process.-Image, 99, 116505, https://doi.org/10.1016/j.image.2021.116505, 2021.

Oliveira, B. B., Carravilla, M. A., and Oliveira, J. F.: A diversity-based genetic algorithm for scenario generation, Eur. J. Oper. Res., 299, 1128–1141, https://doi.org/10.1016/j.ejor.2021.09.047, 2022.

Ravani, B. and Roth, B.: Motion Synthesis using kinematic mappings, ASME J. Mech. Trans. and Automation, 105, 460–467, https://doi.org/10.1115/1.3267382, 1983.

Ruth, D. A. and McCarthy, J. M.: The design of spherical 4R linkages for four specified orientations, Mech. Mach. Theory, 34, 677–692, https://doi.org/10.1016/S0094-114X(98)00048-2, 1999.

Shirazi, K. H.: Computer modelling and geometric construction for four-point synthesis of 4R spherical linkages, Appl. Math. Model., 31, 1874–1888, https://doi.org/10.1016/j.apm.2006.06.013, 2007.

Singh, R., Chaudhary, H., and Singh, A. K.: Defect-free optimal synthesis of crank-rocker linkage using nature-inspired optimization algorithms, Mech. Mach. Theory, 116, 105–122, https://doi.org/10.1016/j.mechmachtheory.2017.05.018, 2017.

Sundram, S. and Larochelle, P.: Using optimization for the mixed exact-approximate synthesis of planar mechanisms, in: ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2–5 August 2015, Boston, Massachusetts, USA, V05BT08A081, https://doi.org/10.1115/DETC2015-47394, 2015.

Sun, J. W. H. and Waldron, K. J.: Graphical transmission angle control in planar linkage synthesis, Mech. Mach. Theory, 16, 385–397, https://doi.org/10.1016/0094-114X(81)90012-4, 1981.

Tipparthi, H. and Larochelle, P.: Orientation order analysis of spherical four-bar mechanisms, J. Mech. Robot., 3, 044501, https://doi.org/10.1115/1.4004898, 2011.

Trejo, O. M., Villar, C. A. C., Escalante, R. P., Arjona, M. A. Z., and Peñuñuri, F.: Synthesis method for the spherical 4R mechanism with minimum center of mass acceleration, Mech. Mach. Theory, 93, 53–64, https://doi.org/10.1016/j.mechmachtheory.2015.04.015, 2015.

Waldron, K. J. and Strong, R. T.: Improved solutions of the branch and order problems of burmester linkage synthesis, Mech. Mach. Theory, 13, 199–207, https://doi.org/10.1016/0094-114X(78)90043-5, 1978.

Wilhelm, S. R., Sullivan, T., and Ven, J. D. V.: Solution rectification of slider-crank mechanisms with transmission angle control, Mech. Mach. Theory, 107, 37–45, https://doi.org/10.1016/j.mechmachtheory.2016.09.011, 2017.

Zhang, S.: Design of Planar Linkage, Higher Education Press, Beijing, China, China Book Number 150100509, 1983.

Zhao, Y.: Analysis and Synthesis of Agricultural Machinery, China Machine Press, Beijing, China, ISBN: 978-7-111-25000-5, 2009.